

upon startup of a second computer of the distributed data processing system, allowing access to the package and user profile by the second computer of the distributed data processing system.

29. (ADDED) A method of managing a distributed data processing system, comprising the computer implemented steps of:

capturing changes in a computer system between an initial state and a modified state;

separating the changes into user-specific changes and system-specific changes;

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

REMARKS

Claims 1-29 are pending in the present application. Reconsideration of the claims is respectfully requested.

Examiner Newgen is thanked for the allowance of Claims 7 and 20. These claims have been amended to include their respective independent claims (as suggested by the Examiner) and their entry is respectfully requested. Claims 7 and 20 are therefore believed in condition for allowance.

I. 35 U.S.C. § 103, Obviousness

The examiner has rejected claims 1, 3-5, 9-11, 13, 14, 16-18, 22-24, and 26-27 under 35 U.S.C. §103 as being unpatentable over Liu in view of Matthews. This rejection is respectfully traversed.

Claim 1 is reproduced for reference:

1. A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

initializing the data processing system for a capture of an initial state of the data processing system;
modifying the data processing system;
capturing a modified state of the data processing system; and
storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes; and
wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

In rejecting Claim 1, the Examiner cites Luu as teaching the claimed limitations of:

storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes...

It is respectfully submitted that Luu does not show the emphasized claim language. The Examiner states in the rejection that Luu specifies system-specific changes and user-specific changes. However, merely mentioning system-specific and user-specific changes does not teach the claimed limitation of, "wherein the differences are separated into system-specific changes and user-specific changes," as claimed in at least Claim 1.

In support of her argument, Examiner cites Luu at col. 6, lines 18-26 and col. 5, lines 25-35. Col. 6 lines 18-26 state:

As noted previously, the preinstallation snapshot contains the state of the system prior to the installation of the application software. The post installation snapshot 502 contains the same information except that it is after the installation of the application software.

This passage discusses the idea of a pre-and post-installation snapshot of system , but does not appear to mention separating the differences into system-specific and user-specific changes.

Col. 5, lines 25-35 of Luu states:

A second file utilized in the present invention is the personality file. The personality file allows for custom installation of application software on a user workstation. For example, if the application is to be installed in a particular directory, it is specified through the personality file. A custom personality file resides on the user workstation. In operation, the installation program on the user workstation will search for a custom personality file. If no custom personality file is found, a default personality file will be utilized to perform the installation.

This passage describes a personality file that resides on the individual workstation. No mention is made of storing the differences between the initial and modified state, wherein the differences are separated into system-specific changes and user-specific changes.

In short, the Examiner has cited two passages, one of which Examiner characterizes as generally showing system-specific changes, the other of which Examiner argues generally shows user-specific changes. However, even if Examiner's characterization of these passages is accepted, the cited reference still does not teach that the differences are separated into system-specific and user-specific changes as claimed in Claim 1.

Claim 1 also includes the language,

wherein the system specific-changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis.

This passage describes an element whose functionality is not shown or taught in Luu. By applying the system-specific changes on a per-system basis and user-specific changes on a per-user basis, Claim 1 describes a system capable of applying system and user profiles independently of one another. This ability provides advantages not present in Luu, as described further below.

According to Luu, no apparent distinction is made between user- and system-specific changes. Though Luu does use a personality profile to help remotely install software, changes to the personality file are not stored anywhere but on the individual terminal, as Luu states at col. 5, lines :

A custom personality file resides on the user workstation. In operation, the installation program on the user workstation will search for a custom personality file. If no custom personality file is found, a default personality file will be utilized to perform the installation.

First, note that this passage states Luu's personality file is used to remotely install the new software, i.e., it is part of the tool making the changes as taught by Luu, not part of the measured changes themselves as in the present application. Thus, the personality profile of Luu is not equal to the user-specific changes of the present application, because Luu uses the existing personality file to install software, while the present invention captures and stores any user-specific changes.

Second, Luu makes no mention of recording changes to its personality file, as embodied by the claim language in Claim 1:

capturing a modified state of the data processing system....storing differences between the initial state and the modified state...wherein the differences are separated into system-specific changes and user-specific changes...

In order for Luu's personality file to teach the user-specific changes in Claim 1, the personality file would need to (1) have an initial and a modified state (no such states are taught in Luu) and (2) have the differences between those states applied on a per-user basis. No such teaching is found in Luu.

Therefore, it is respectfully urged that Claim 1 is distinguished over the cited references.

Furthermore, independent Claims 14 and 27, rejected under the same rationale as Claim 1, are also believed distinguished from the cited references by virtue of the above arguments.

Likewise, all independent Claims are, by virtue of the foregoing arguments, believed distinguished from the cited references.

Therefore, the rejection of all under 35 U.S.C. § 103 has been overcome.

II. Objection to Claims

The Examiner has stated that claims 7 and 20 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. In response, the claims have been rewritten to overcome this objection, as indicated above.

III. Restriction Requirement

Applicant respectfully traverses the requirement for restriction of Claim 28 from the present application.

The MPEP states at sec. 803.01: CRITERIA FOR RESTRICTION BETWEEN PATENTABLY DISTINCT INVENTIONS

There are two criteria for a proper requirement for restriction between patentably distinct inventions:

- (A) The inventions must be independent...or distinct as claimed...; and
- (B) There must be a serious burden on the Examiner if restriction is required...

The MMPEP also states, at section 803.01: GUIDELINES:

For purposes of the initial requirement, a serious burden on the Examiner may be *prima facie* shown if the Examiner shows by appropriate explanation of separate classification, or separate status in the art, or a different field of search as defined in MPEP sec. 808.02.

In this case, the Examiner has restricted Claim 28 without such showing, only stating,

Newly submitted Claim 28 is directed to an invention that is independent or distinct from the invention originally claimed for the following reasons: creating a package file, creating a user profile, storing the package and user profile, allowing access to the package and user profile by the second computer.

It is respectfully submitted that this statement does not meet the required *prima facie* case to impose a restriction requirement on Claim 28 of the present application. No argument has been presented as to why the limitations of Claim 28 are independent or distinct, and no showing has been made of a serious burden on the Examiner. Therefore, the restriction requirement is respectfully traversed.

Furthermore, Claim 28 is believed distinguished from the cited references, and is believed to be in condition for allowance.

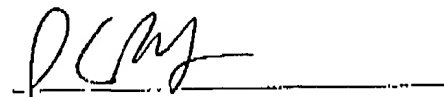
IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 3-10-03

Respectfully submitted,



Patrick C. R. Holmes
Reg. No. 46,380
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

REDACTED CLAIMS

7. (ONCE AMENDED) [The method of 6] A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

initializing the data processing system for a capture of an initial state of the data processing system;

modifying the data processing system;

capturing a modified state of the data processing system; and

storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes; [and]

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis;

wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries; and

wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.

20. (ONCE AMENDED) [The apparatus of 19] An apparatus for identifying and storing changes to a data processing system within a distributed data processing system, the apparatus comprising:

initializing means for initializing the data processing system for a capture of an initial state of the data processing system;

modifying means for modifying the data processing system;

capturing means for capturing a modified state of the data processing system; and

storing means for storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes; [and]

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis;

wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries; and

wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.